COURSE TITLE

Coding 201

LENGTH

One Semester

DEPARTMENT

Computer Technology
Barbara O'Donnell, Supervisor

SCHOOL

Union Middle School

DATE

Spring 2017

Coding 201

I. Introduction/Overview/Philosophy

In this course, students will explore a programing language(s) more deeply. The students will explore programming concepts and develop programs with graphics, animation, and interactivity. Game design will also be explored. Students must have successfully completed Computer Coding 1 to enroll in this course.

II. Objectives

Students will:

- Add backgrounds, sprites and shapes
- Use action commands and events to create animated and interactive projects
- Learn the basic concepts of computer programming (coding), including:
  - Variables
  - Data types
  - Debugging
  - Loops
  - If-statements
  - Nesting
- Extend their knowledge of coding to include more advanced topics, such as:
  - Functions
  - Parameters
  - Returning Values
  - Programming logic
  - Dictionaries
  - Events
- Learn to create and customize popular apps and classic video games.

Course Outline

A. Program Design and Implementation
   a. Algorithms in action
   b. Learn syntax

B. Variables
   a. What is a variable?
   b. How to create a new variable
   c. How to change and copy variables

C. Math and coding
   a. What are expressions and operators?
   b. How to calculate sums

c. Using brackets to ensure accuracy

D. Strings and inputs
   a. What is a string?
   b. Adding strings
   c. Getting strings as input from the keyboard

E. Loops
   a. What are loops?
   b. All about for loops
   c. Using ranges

F. Decisions
   a. Boolean expressions (true or false) and operators (and, or)
   b. What is a branch in programming?
      i. If and If-Else
      ii. More advanced branching
   c. While loops
      i. What is a while loop?
      ii. How to use a while loop
      iii. How to make a while loop go on forever
      iv. How to escape from a while loop that won't stop

G. Functions
   a. What is a function?
   b. How to create and use functions

H. Intro to Game Design
   a. Create and customize popular apps and classic video games
   b. Design and edit gaming programs

I. Career Exploration
   a. Research and explore a variety of computer science related careers
   b. Understand how computer science impacts all careers

New Jersey Student Learning Standards

TECHNOLOGY

Standard 8.1: Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaboratively and to create and communicate knowledge.
Standard 8.2: Technology Education, Engineering, and Design: All students will develop an understanding of the nature and impact of technology, engineering, technological design, and the designed world, as they relate to the individual, global society, and the environment.

Strand E. Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.

21ST CENTURY LIFE AND CAREERS

Standard 9.2: Career Awareness, Exploration, and Preparation
Standard 9.3 – Career & Technical Education (CTE)
Pathway: Programming & Software Development (IT□PRG)

III. Proficiency Levels

This course is open to grades 7 and 8 who have successfully completed Coding 101.

IV. Methods of Assessment

Student Assessment
The teacher will provide a variety of assessments during the course of the year. Among these are: homework, laboratory exercises, weekly projects, teacher-made tests and quizzes, and long-term projects.

Curriculum/Teacher Assessment
The teacher will provide the subject area supervisor with suggestions for changes on an ongoing basis.

V. Grouping

This is a middle school elective course offered to students in grades 7 and 8.

VI. Articulation/Scope & Sequence/Time Frame

Course length is one semester.

VII. Resources

Resources include but are not limited to:

- **Computer Coding** by Jon Woodcock, DK Workbooks, 2014.
- **How to Code** by Max Wainewright, QEB Publishing, 2015.
- **Beyond the Hour of Code** http://www.beyondthehourofcode.com/
- http://cty.jhu.edu/ctyonline/courses/computer_science/scratch_programming.html
- https://www.khanacademy.org/computing/computer-programming/programming-games-visualizations
- http://www.bootstrapworld.org/

VIII. Methodologies

The following methods of instruction are suggested: lecture, group projects, demonstration, hands-on applications, and class presentations.

IX. Suggested Activities

- Laboratory programming problems
- Game simulated programs
- Cooperative programming projects

X. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas. Connections are also made to the disciplines of business, art and English, by means of incorporation of these ideas into programming projects.

XI. Differentiating Instruction for Students with Special Needs: Students with Disabilities, English Language Learners, and Gifted & Talented Students

Differentiating instruction is a flexible process that includes the planning and design of instruction, how that instruction is delivered, and how student progress is measured. Teachers recognize that students can learn in multiple ways as they celebrate students' prior knowledge. By providing appropriately challenging learning, teachers can maximize success for all students.

Examples of Strategies and Practices that Support:

Students with Disabilities

- Use of visual and multisensory formats
- Use of assisted technology
- Use of prompts
- Modification of content and student products
- Testing accommodations
- Authentic assessments

English Language Learners

- Pre-teaching of vocabulary and concepts
- Visual learning, including graphic organizers
- Use of cognates to increase comprehension
- Teacher modeling
- Pairing students with beginning English language skills with students who have more advanced English language skills
- Scaffolding
    - word walls
    - sentence frames

- think-pair-share
- cooperative learning groups

Gifted & Talented Students

- Adjusting the pace of lessons
- Curriculum compacting
- Inquiry-based instruction
- Independent study
- Higher-order thinking skills
- Interest-based content
- Student-driven
- Real-world problems and scenarios

XII. Professional Development

The teacher will continue to improve expertise through participation in a variety of professional development opportunities.

Curriculum Map

Week 1	Week 2	Week 3	Week 4	Week 5
Getting started programming		Basic Concepts: Variables Data Types Debugging Loops If Statements Nesting		

Week 6	Week 7	Week 8	Week 9	Week 10
Basic Concepts: Variables Data Types Debugging Loops If Statements Nesting			Advanced Topics: Functions Parameters Returning Values Programming Logic Dictionaries Events	

Week 11	Week 12	Week 13	Week 14	Week 15
Advanced Topics: Functions Parameters Returning Values Programming Logic Dictionaries Events				Culminating Project

Week 16	Week 17	Week 18	Week 19	Week 20
Culminating Project	Game Design		Career Exploration	